

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07226736 A**

(43) Date of publication of application: **22 . 08 . 95**

(51) Int. Cl. **H04L 12/00**
H04L 12/42
H04L 12/24
H04L 12/26
H04L 29/14

(21) Application number: **06017110**

(22) Date of filing: **14 . 02 . 94**

(71) Applicant: **HITACHI LTD**

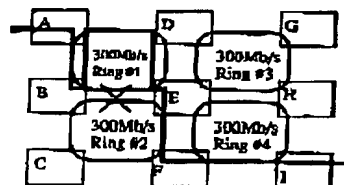
(72) Inventor: **TAKATORI MASAHIRO**
NAKANO YUKIO
ASHI MASAHIRO

(54) **FAULT RECOVERY METHOD IN MESH NETWORK** COPYRIGHT: (C)1995,JPO

(57) Abstract:

PURPOSE: To obtain a restoration method with high reliability at a high speed by dividing logically a network into plural ring networks and a fault detection node communicates signaling over a closed range in the inside of the logical ring network.

CONSTITUTION: When a fault takes place in a transmission line between nodes B and E, fault end nodes B, E make a fault recovery request and only nodes to be active are those belonging to logic rings #1, #2. When a VC-4 path comprises a path of A-B-E-F-I, the nodes B, E exchange a message and a bypass with respect to a traffic between the nodes B-E is decided independently by the logic rings #1, #2. The bypass of the traffic belonging to the logic ring #1 is B-A-D-E and the bypass of the traffic belonging to the logic ring #2 is B-C-F-E. In the case of the path above, since the path belongs to the logic ring #1 between the nodes B and E, a path after fault recovery is A-B-A-D-E-F-I. An active VC is used between the nodes A and B and among the nodes E-F-I and a standby VC is used among the nodes B-A-D-E.



Printed from Mimosa

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-226736

(43) 公開日 平成7年 (1995) 8月22日

(51) Int. Cl. ⁶

H04L 12/00

12/42

12/24

識別記号

庁内整理番号

F I

技術表示箇所

8732-5K

H04L 11/00

8838-5K

330

審査請求 未請求 請求項の数17 (全 13 頁) 最終頁に続く

(21) 出願番号 特願平6-17110

(22) 出願日 平成6年 (1994) 2月14日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 ▲高▼取 正浩

東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

(72) 発明者 中野 幸男

東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

(72) 発明者 芦 賢浩

神奈川県横浜市戸塚区戸塚町216番地 株式会社日立製作所情報通信事業部内

(74) 代理人 弁理士 小川 勝男

(54) 【発明の名称】 メッシュ網における障害復旧方法

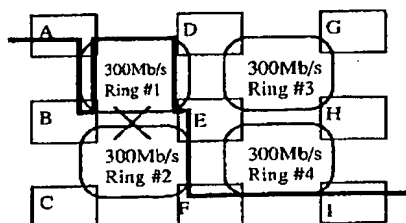
(57) 【要約】

【目的】 メッシュ状のネットワークにおいて、障害発生時に自動的に復旧動作可能な網構成方法と復旧アルゴリズムを提供する。

【構成】 ネットワークの閉ループごとに論理リング1～4を設定し、障害発生時には、論理リングごとに該論理リング内でトラヒックを障害発生方向とは逆方向に迂回させ、復旧する。

【効果】 本発明によれば、ネットワークを複数の論理リングに分割し、迂回路の設定をおこなうため、障害復旧にかかる時間が小さくなるという効果がある。

図1



【特許請求の範囲】

【請求項1】メッシュ状に接続されたネットワーク内部の閉ループを論理リングと定義し、伝送路内部に多重化されている複数のチャンネルごとに所属する論理リングを予め定めておき、前記論理リングにおいては、当該論理リングに所属するチャンネルを二分し、一方を現用チャンネル、他方を予備チャンネルとするネットワークにおけるトラヒックの救済方法であって、前記救済するトラヒックが伝送されている方向とは反対まわりの方向の当該論理リング内部の予備チャンネルを用いて、送信側障害端ノードから受信側障害端ノードまで当該トラヒックを伝達することにより、障害から復旧することの特徴とするメッシュ網における障害復旧方法。

【請求項2】メッシュ状に接続されたネットワーク内部の閉ループを論理リングと定義し、伝送路内部に多重化されている複数のチャンネルごとに属する論理リングを予め定めておき、前記論理リングにおいては、当該論理リングに所属するチャンネルを二分し、一方を現用チャンネル、他方を予備チャンネルとするネットワークにおけるトラヒックの救済方法であって、前記論理リングに障害が発生した場合で、かつ、当該障害により当該論理リングからノードの切り離しが発生しない場合、または、ノードの切り離しが発生した場合においても救済するトラヒックが前記切り離されたノードにおいて当該論理リング内の他のノードに接続される場合、前記救済するトラヒックが伝送されている方向とは反対まわりの方向の当該論理リング内部の予備チャンネルを用いて、送信側障害端ノードから受信側障害端ノードまで当該トラヒックを伝達することにより、障害から復旧することの特徴とするメッシュ網における障害復旧方法。

【請求項3】メッシュ状に接続されたネットワーク内部の閉ループを論理リングと定義し、伝送路内部に多重化されている複数のチャンネルごとに属する論理リングを予め定めておき、前記論理リングにおいては、当該論理リングに所属するチャンネルを二分し、一方を現用チャンネル、他方を予備チャンネルとするネットワークにおけるトラヒックの救済方法であって、前記論理リングに障害が発生した場合で、かつ、当該障害により当該論理リング（論理リングA）からノードの切り離しが生じる場合で、かつ、救済するトラヒックが前記切り離されたノードにおいて他の論理リング（論理リングB）に接続される場合で、かつ、当該論理リングAにおいて前記救済するトラヒックが通過する障害端ノードが前記論理リングBにも属している場合、前記論理リングAに属する障害端ノードから論理リングBの予備チャンネルに当該トラヒックを接続し、論理リングBに属する障害端ノードまで当該トラヒックを伝達することにより、障害から復旧することの特徴とするメッシュ網における障害復旧方法。

【請求項4】メッシュ状に接続されたネットワーク内部

の閉ループを論理リングと定義し、伝送路内部に多重化されている複数のチャンネルごとに属する論理リングを予め定めておき、前記論理リングにおいては、当該論理リングに所属するチャンネルを二分し、一方を現用チャンネル、他方を予備チャンネルとするネットワークにおけるトラヒックの救済方法であって、前記論理リングに障害が発生した場合で、かつ、当該障害により当該論理リング（論理リングA）からノードの切り離しが生じる場合で、かつ、障害により影響をうけるトラヒックが前記切り離されたノードにおいて他の論理リング（論理リングB）に接続される場合で、かつ、当該論理リングAの障害端ノードが前記論理リングBに属していない場合、前記論理リングAに属する障害端ノードから論理リングAの予備チャンネルに当該トラヒックを接続し、論理リングAとBの両方に属する他の障害端ノードを経由して、論理リングBに属するもう一方の障害端ノードまで当該トラヒックを伝達することにより、障害から復旧することの特徴とするメッシュ網における障害復旧方法。

20 【請求項5】前記論理リング毎に各ノードが互いにメッセージを交換し、障害情報を伝達することにより自律分散的に処理を実行することの特徴とする請求項1乃至4のいずれかに記載のメッシュ網における障害復旧方法。

【請求項6】前記論理リングに属するチャンネルに番号（論理チャンネル番号）を付与し、ノードの入力回線と出力回線内のチャンネルが当該ノードの属する論理リングにおいて、どの論理チャンネル番号を付与されているかを示す情報を保持していることの特徴とする請求項5に記載のメッシュ網における障害復旧方法。

30 【請求項7】前記各ノードは、所属している論理リング内の全てのノード番号の情報およびノードの接続順序情報を保持していることの特徴とする請求項5に記載のメッシュ網における障害復旧方法。

【請求項8】前記各ノードは、所属している論理リング内の全てのトラヒックの始点ノード番号および終点ノード番号情報、および該トラヒックが隣接する論理リングに接続されている場合にはその隣接する論理リング番号情報および該隣接する論理リングにおいて当該トラヒックに与えられている論理チャンネル番号を保持すること40 を特徴とする請求項5に記載のメッシュ網における障害復旧方法。

【請求項9】前記各ノードは、所属している論理リング番号情報を保持することの特徴とする請求項5に記載のメッシュ網における障害復旧方法。

【請求項10】前記メッセージは、相手ノード番号、送信ノード番号、切替命令情報、送信ノードの切替状態を情報としてもつことの特徴とする請求項5に記載のメッシュ網における障害復旧方法。

50 【請求項11】前記メッセージは、前記内容に加えて論理リング番号を含むことの特徴とする請求項10に記載

のメッシュ網における障害復旧方法。

【請求項12】前記論理リングが複数存在する場合、1つのノードだけでしか接していない論理リング間でのトラヒックの接続を行わないことを特徴とする請求項5に記載のメッシュ網における障害復旧方法。

【請求項13】前記論理リングの2つにまたがるようにトラヒックを設定する場合、前記2つの論理リングの両方に所属するノードを少なくとも2つは経由することを特徴とする請求項5に記載のメッシュ網における障害復旧方法。

【請求項14】前記ネットワークには、CCITT勧告G.707, 708, 709に定めるSDH(Synchronous Digital Hierarchy)、もしくはANSI T1.105に定めるSONET(Synchronous Optical Network)を用いることを特徴とする請求項1乃至5のいずれかに記載のメッシュ網における障害復旧方法。

【請求項15】前記チャンネルは、CCITT勧告G.707, 708, 709に定めるVC(Virtual Container)、もしくはANSI T1.105に定めるSTS-N SPE(Synchronous Transport Signal level N Synchronous Payload Envelope)であることを特徴とする請求項14に記載のメッシュ網における障害復旧方法。

【請求項16】前記ノード間において用いるメッセージの伝達にはバスオーバーヘッドを用いることを特徴とする請求項14に記載のメッシュ網における障害復旧方法。

【請求項17】前記ノード間において用いるメッセージの伝達にはセクションオーバーヘッド内のデータコミュニケーションチャンネル(DCC)を用いることを特徴とする請求項14に記載のメッシュ網における障害復旧方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、伝送路で接続される複数のノードを有するネットワークにおいて、ノードや伝送路に障害が発生した場合に自動的にトラヒックを迂回させ、復旧を行うメッシュ網における障害復旧方法に関する。

【0002】

【従来の技術】従来は、例えば、「Bellcore TA-NWT-001230, Issue 3」に示されるように、単一のリング型ネットワークにおいて障害発生時に伝送路上に伝送されていたトラヒックを伝送路(ライン)もしくはバスを反対方向に折り返すことにより復旧する方法がある(従来例1)。これは、複数ノードをリング状に接続し、各ノード間の容量を等しくし、かつ、該容量を2等分し、一方を現用、他方を予備として用いるリング状ネットワークにおいて、通常は現用容量のみを用いてトラヒックを伝送し、障害発生時に予備容量を用いてトラヒックを通常時とは反対方向に折り返すことにより、迂回復旧する方

法である。すなわち、障害を検出したノードは、障害発生方向に出力していたトラヒックを反対方向の予備回線を用いて折り返す。(以下、ブリッジと称する。)また、通常時障害発生箇所の現用回線から受信していたトラヒックを反対方向の予備回線から受信し、現用回線に接続する。(以下、スイッチと称する。)上記動作により単一リング型ネットワークにおいて障害から復旧する。

【0003】あるいは、「GLOBECOM'87 28.2 THE SELFHEALING NETWORK」に示されるように、メッシュ網において障害検出ノードがメッセージを交換し、迂回路を探索し、復旧する方法もある(従来例2)。この方法では、障害端ノードのうち一方が、隣接するすべてのノードに対してメッセージを送出する。メッセージの内容は、1)送信ノード番号、2)相手ノード番号、3)復旧に要する容量などである。このメッセージを受信したノードは、空き容量を考慮し、復旧に必要な容量を確保できるすべての出力回線に該メッセージを同報出力する。このような制御により、メッセージは複数の経路を通過し、もう一方の障害端ノードに到着する。迂回路としては、例えば、最も速く到着したメッセージが通過した経路を採用する。上記方式により、メッシュ型ネットワークにおいて障害から復旧する。

【0004】

【発明が解決しようとする課題】しかし、上記の従来例1における方法は、メッシュ状のネットワークに適用することはできず、応用できる範囲がせまい。また、上記の従来例2においては、メッシュ状のネットワークが考慮されており、障害に対する柔軟性ももっているが、処理に時間がかかる。また、ロバストネスに欠けるとい

点ももっている。

【0005】本発明の目的は、メッシュ状のネットワークにおいて障害を検出した場合に、迅速かつ信頼性の高い復旧方法を提供することにある。

【0006】

【課題を解決するための手段】本発明は上記目的を達成するために、網を複数のリング網に論理的に分割し、論理リング網内部に閉じた範囲で障害検出ノードがシグナリングをかわすことにより復旧処理を実行することとしたものである。

【0007】

【作用】本発明の作用を図2を用いて説明する。図2は、図3に示す9ノードからなるネットワークを論理的に4つのリング網に分割した様子を示している。

【0008】各論理リングのもつ容量は、現用と予備に2分されており、通常は現用帯域のみを用いて伝送をおこなう。そして、障害発生時には、障害によって影響を受けたトラヒックを障害発生方向と反対回りに伝送することにより、論理リング内においてに閉じた範囲で迂回路を形成し、障害から復旧する。

【0009】例えば、図2においてノードAとノードIの間で伝送を行う場合、経路をA-B-E-F-Iの順に設定する。これを図4に示す。そして、ノードA-B-E間ではトラヒックは論理リング1に属し、ノードE-F間では論理リング2に属し、ノードF-I間では論理リング4に属するように設定する。障害復旧の例として、ノードB-E間の伝送路に障害が発生した場合をとりあげる。この場合、障害復旧要求を出すのは障害端ノードBとEであり、復旧動作を行うのは、論理リング1と2に属するノードだけである。上記の経路をとる場合、ノードBとノードEがメッセージを交換し、ノードB-E間のトラヒックに対する迂回路を論理リング1と2で決定する。論理リング1に属するトラヒックの迂回路はB-A-D-Eであり、論理リング2に属するトラヒックの迂回路はB-C-F-Eである。上記のトラヒックの場合、ノードB-E間においては論理リング1に属しているので、障害復旧後の経路は図1に示すように、A-B-A-D-E-F-Iとなる。

【0010】なお、ノードAとノードIの間で伝送を行う場合、ノードEにおいて論理リング1から論理リング4へトラヒックを接続するような経路設定は、ノードEに障害が発生した場合に該トラヒックを救済できないのであらかじめ禁止する。

【0011】

【実施例】本発明の第1の実施例を図2を用いて説明する。図2におけるネットワークは、図3に示す9ノードを格子状に接続したネットワークにおいて、内部の4つの閉ループに着目し、これらを論理的に4つのリング網と見なしたものである。ここで、4つの論理リングを各々論理リング1~4とする。各ノード間の回線容量は、ノードB-E間、ノードE-H間、ノードD-E間、ノードE-F間は2.48832Gb/sであり、その他は622.08Mb/sである。回線は双方向通信である。

【0012】本発明の説明に用いる伝送方式は、CCITT勧告G.707, 708, 709に定められているSynchronous Digital Hierarchy(SDH)である。前記622.08Mb/s回線で用いるフレームは、Synchronous Transport Module level 4(STM-4)と呼ばれるものであり、そのフレームフォーマットを図5に示す。また、前記2.48832Gb/s回線で用いるフレームは、STM-16と呼ばれるものであり、そのフレームフォーマットを図6に示す。いずれも伝送用のペイロードとしてはVirtual Container level 4(VC-4)を用いる。STM-4は4つのVC-4、STM-16は16のVC-4を収容している。

【0013】ここでネットワーク内の各論理リングの容量は622.08Mb/sとする。これら論理リングにおいては、その容量を二分し、一方を現用、他方を予備として用いる。そして、通常時は現用のみを用いて伝送を行い、障害発生時には予備を用いて伝送を行う。本例の場合は、容量622.08 Mb/sを311.04 Mb/sずつに分ける。すなわち、各論理リングの現用容量は、311.04 Mb/sとなる。

【0014】ここで各論理リングにおいて論理VCを定義する。論理VCとは、論理リング内において各伝送路内部のVCの属するグループである。各論理リングにおける論理VC番号と各回線のVC番号との対応を図7に示す。例えば、ノードB-E間において、回線内のVC-4 #1から#4は論理リング1に属し、VC-4 #5から#8までは論理リング2に属する。論理リング1において、VC-4 #1, #2は現用であり、VC-4 #3, #4は予備である。論理リング2において、VC-4 #5, #6は現用であり、VC-4 #7, #8は予備である。この場合、論理リング1の現用論理VC#1(論理VC#1w)としてVC-4 #1を、予備論理VC#1(論理VC#1p)としてVC-4 #3を登録する。また、論理リング2の現用論理VC#1(論理VC#1w)としてVC-4 #5を、予備論理VC#1(論理VC#1p)としてVC-4 #7を登録する。他の回線についても同様に登録をする。論理VC番号とVC番号を一致させる必要はない。論理リング内部では、論理VC番号を用いてパスを設定する。

【0015】次に障害時における復旧手法について説明する。いま、信号の伝達経路は図4に示すように設定されているものとする。すなわち、VC-4パスがA-B-E-F-Iの経路を通過する。ここで、当該パスは、ノードA-B-E間では論理リング1に属し、論理VC番号は1に設定されている。ノードE-F間では論理リング2に属し、論理VC番号は2である。ノードF-I間では論理リング4に属し、論理VC番号は1である。障害復旧の例として、ノードB-E間の伝送路に障害が発生した場合をとりあげる。この場合、障害復旧要求を出すのは障害端ノードBとEであり、動作を行うのは、論理リング1と2に属するノードだけである。上記の経路をとる場合、ノードBとノードEがメッセージを交換し、ノードB-E間のトラヒックに対する迂回路を論理リング1と2で独立に決定する。論理リング1に属するトラヒックの迂回路はB-A-D-Eであり、論理リング2に属するトラヒックの迂回路はB-C-F-Eである。上記のパスの場合、ノードB-E間においては論理リング1に属しているので、障害復旧後の経路は図1に示すように、A-B-A-D-E-F-Iとなる。このうち、A-B間とE-F-I間は現用VC、B-A-D-E間は予備VCを用いて伝送される。

【0016】上記処理における各ノードの動作について詳細に述べる。ここで各ノードの構成は同じとする。この場合、ノードEは信号断を検出し、ノードBに対してメッセージを送出する。ノード間のシグナリングについては、Bellcore TA-NWT-001230, Issue 3とほぼ同じものを用いて説明する。

【0017】メッセージはM1, M2の2種類からなる。その内容を図8に示す。M1の第1-4ビットは、切替命令種類および優先度を示す。これを表1に示す。

【0018】

【表1】

10

20

30

40

切替要求の優先順位 (表1)

Bits 1-4	切替要求名称	略称
1111	Lockout of Protection or Signal Fail(Protection)	LP-P SF-P
1110	Forced Switch (Span)	FS-S
1101	Forced Switch (Ring)	FS-R
1100	Signal Fail (Span)	SF-S
1011	Signal Fail (Ring)	SF-R
1010	Signal Degrade(Protection)	SD-P
1001	Signal Degrade (Span)	SD-S
1000	Signal Degrade (Ring)	SD-R
0111	Manual Switch (Span)	MS-S
0110	Manual Switch (Ring)	MS-R
0101	Wait-To-Restore	WTR
0100	Exerciser (Span)	EXER-S
0011	Exerciser (Ring)	EXER-R
0010	Reverse Request(Span)	RR-S
0001	Reverse Request(Ring)	RR-R
0000	No Request	NR

【0019】これらは、「Bellcore TA-NWT-001230, Issue 3」に示されている単一リング型ネットワークにおける障害の種類および優先度を示すコードと一致している。表において、LP-Sとは、あるノード間の回線（以下、spanと称する。）の予備容量の使用を禁止する命令である。FS-Sとは、現用回線を予備回線に強制的に切り替える命令である。FS-Rとは、論理リングの現用容量内を通過するトラヒックを論理リング内の反対回り方向の予備容量を通過するように強制的に折り返す命令である。SF-Sとは、信号断をトリガとし、自動的に現用回線を予備回線に切り替えるための命令である。SF-Rとは、信号断をトリガとし、自動的に論理リングの現用容量内を通過するトラヒックを論理リング内の反対回り方向の予備容量を通過するように折り返すための命令である。SD-Pとは、予備回線に信号誤り率劣化を検出した場合に、当該予備回線を使用禁止にするための命令である。SD-Sとは、信号誤り率劣化をトリガとし、自動的に現用回線を予備回線に切り替えるための命令である。SD-Rとは、信号誤り率劣化をトリガとし、自動的に現用容量内を通過するトラヒックを反対回り方向の予備容量を通過するように折り返すための命令である。MS-Sとは、現用回線を予備回線に強制的に切り替える命令である。FS-S

メッセージの種類 (表2)

M2 bit5 code	M1 bits 1-4 code	
	Ring Bridge code	Span Bridge code
Long Path	Request	Status
Short Path	Status	Request

にくらべ優先度が低い。MS-Rとは、現用容量内を通過するトラヒックを反対回り方向の予備容量を通過するように強制的に折り返す命令である。FS-Rにくらべ優先度が低い。WTRとは、障害が消滅した場合に一定時間切り戻しを禁止する命令である。EXER-Sとは、現用回線を予備回線に強制的に切り替えるためのシグナリングを出力する命令で、ブリッジおよびスイッチは実行しない。MS-Sにくらべ優先度は低い。EXER-Rとは、現用容量内を通過するトラヒックを反対回り方向の予備容量を通過するように強制的に折り返すためのシグナリングを出力する命令で、ブリッジおよびスイッチは実行しない。MS-Rにくらべ優先度が低い。RR-Sとは、FS-S、SF-S、SD-S、MS-S、EXER-Sを受信したノードが出力する応答(ACK)である。RR-Rとは、FS-R、SF-R、SD-R、MS-R、EXER-Rを受信したノードが出力する応答(ACK)である。NRとは、切替要求のない場合に出力する命令である。それぞれに優先度が定められており、表の上に行くほど優先度が高い。

【0020】次に、M1の第5-8ビットは、相手ノードの番号を示す。

20 【0021】次に、M2の第1-4ビットは、該メッセージを発生したノードの番号を示す。M2の第5ビットにより、メッセージがshort pathに出されたものか、long pathに出されたものかを示す。ここでshort pathとは障害の発生している区間であり、long pathとは、論理リングにおいて障害の発生している区間以外をさす。すなわち本例の論理リング1においては、B-Eの区間がshort pathであり、B-A-D-Eの区間がlong pathとなる。また、論理リング2においては、B-Eの区間がshort pathであり、B-C-F-Eの区間がlong pathとなる。M2のbit5が'1'であれば、当該メッセージがLong Pathに出力されたものであることを示し、M2のbit5が'0'であれば、当該メッセージがShort Pathに出力されたものであることを示す。受信側では、該メッセージを受信した方向と受信M2のbit5の値の組み合わせによって、受信メッセージがRequest情報なのかStatus情報なのかを判断する。これを表2に示す。

【0022】

【表2】

【0023】次に、M2の第6-8ビットはノードの状態を示す。これを表3に示す。

【0024】

【表3】

M2 Bit-8のStatus code (表3)

M2 Bit-8	Status
1 1 1	MS-AIS
1 1 0	MS-FERF
1 0 1	Reserved for future use
1 0 0	Reserved for future use
0 1 1	Reserved for future use
0 1 0	Bridged and Switched
0 0 1	Bridged
0 0 0	Idle

【0025】 Bridged (Br)は該メッセージを送信したノードにおいてブリッジが完了したことを示し、Bridged and Switched (以下、Br&Sw)は該メッセージを送信したノードにおいてブリッジとスイッチが完了したことを示す。また、IDLEとはノードにおいてBridgeもSwitchも行っていないことを示す。これらメッセージの内容は実際は2進数で表現されるが、ここでは説明を容易にするため、表1～3に示す表記方法を用いる。以下、M1、M2の内容を示す場合、

M1=(切替優先度)/(相手ノード番号)

M2=(自ノード番号)/(Short or Long)/(状態)

のように表現する。

【0026】 本例において、ノードEは障害を検出し、short path方向にM1=SF-R/B、M2=E/S/FERF、long path方向にM1=SF-R/B、M2=E/L/IDLEを送出する。このメッセージの伝達には、論理リング内の2つの予備VC-4のPOH(Path Overhead)のZ5バイトを用いる。すなわち、VC-4 #3のZ5バイトによりM1を、VC-4 #4のZ5バイトによりM2を伝達する。

【0027】 long pathに存在するノードDは、ノードEから送出されたメッセージを受信する。そしてM1を調べ、当該メッセージが自分宛でないのを検出し、当該メッセージを論理リング1内の次段（この場合はノードA）に伝送する。また、論理リング1において、受信側予備論理VCも送信側予備論理VCに接続する。同様の処理をノードAでも実行する。

【0028】 ノードBは、ノードEの出力したメッセージのうち、まずshort path方向からのメッセージ (M1=SF-R/B、M2=E/S/FERF)を受信する。これによりノードBはB-E間に障害が発生したことを知り、short path方向にM1=RR-R/E、M2=B/S/IDLE、long path方向にM1=SF-R/E、M2=B/L/IDLEを送出する。

【0029】 次にノードBは、ノードEの出力したlong path方向からのメッセージ (M1=SF-R/B、M2=E/L/IDLE)を受信する。これにより迂回路から正常な切替要求を受信するので、ノードBはブリッジとスイッチを実行する。つまり、いままでノードE方向に送出していたトラヒックをノードAの方向の予備論理VCにブリッジする。すなわち、ノードB-E間の論理VC-4 #1、#2をノードB-A間の論

理VC-4 #3、#4にブリッジする。あわせてノードEから受信していた論理VC-4 #1、#2をノードAから受信する論理VC-4 #3、#4にスイッチする。そしてshort path方向にM1=RR-R/E、M2=B/S/Br&Sw、long path方向にM1=SF-R/E、M2=B/L/Br&Swを送出する。

【0030】 ノードAは、ノードBから送出されたメッセージを受信する。そしてM1を調べ、当該メッセージが自分宛でないのを検出し、当該メッセージを論理リング1内の次段（この場合はノードD）に伝送する。同様の処理をノードDでも実行する。

【0031】 ノードEは、ノードBがlong path方向に送出したメッセージ (M1=SF-R/B、M2=E/L/IDLE)を受信する。これにより迂回路から正常な切替要求を受信し、いままでノードB方向に送出していたトラヒックをノードDの方向の予備論理VCにブリッジする。すなわち、ノードB-E間の論理VC-4 #1、#2をノードE-D間の論理VC-4 #3、#4にブリッジする。あわせてノードBから受信していた論理VC-4 #1、#2をノードDから受信する論理VC-4 #3、#4にスイッチする。そしてshort path方向にM1=RR-R/B、M2=E/S/FERF、long path方向にM1=SF-R/B、M2=E/L/Br&Swを送出する。

【0032】 上記シグナリングにより、ノードBとEの間でメッセージを交換し、図1に示すような迂回路を形成する。図9に上記メッセージの論理リング1における伝達シーケンスを示す。また図10に論理リング2における伝達シーケンスを示す。このようにして形成された迂回路を用いて、前記パスはA-B-A-D-E-F-Iの経路を通過することになる。このうち、A-B間は論理リング1の現用、B-A-D-E間は論理リング1の予備、E-F間は論理リング2の現用、F-I間は論理リング4の予備論理VCを用いる。

【0033】 次にノードBに障害が発生した場合の処理について述べる。信号の伝達経路は、この場合も図4に示すように設定されているものとする。本例においてノードBに障害が発生した場合、論理リング1においては、ノードAとEが障害を検出する。論理リング2においてはノードCとEが障害を検出する。したがって、ノードBに障害が発生した場合、障害復旧要求を出すのは論理リング1では、障害端ノードAとEであり、論理リング2では、障害端ノードCとEである。論理リング1では、ノードAとノードEがメッセージを交換し、ノードA-B間とB-E間のトラヒックに対する迂回路をA-D-E間の予備容量を用いて構成する。論理リング2では、ノードCとノードEがメッセージを交換し、ノードC-B間とB-E間のトラヒックに対する迂回路をC-F-E間の予備容量を用いて構成する。図4のパスの場合、ノードA-B-E間においては論理リング1に属しているので、障害復旧後の経路は図11に示すように、A-D-E-F-Iとなる。このうち、A-D-E間は予備VC、E-F-I間は現用VCを用いて伝送される。

【0034】 上記処理における各ノードの動作について

詳細に述べる。障害発生後、ノードEは障害を検出し、short path方向にM1=SF-R/B、M2=E/S/FERF、long path方向にM1=SF-R/B、M2=E/L/IDLEを送出する。また、ノードAも障害を検出し、short path方向にM1=SF-R/B、M2=A/S/FERF、long path方向にM1=SF-R/B、M2=A/L/IDLEを送出する。

【0035】long pathに存在するノードDは、ノードEとAから送出されたメッセージを受信する。そしてM1を調べ当該メッセージが自分宛でないのを検出し、それぞれ次段に中継する。また、論理リング1において、受信側予備論理VCも送信側予備論理VCに接続する。

【0036】次にノードAは、ノードEの送出したlong path方向からのメッセージ (M1=SF-R/B、M2=E/L/IDLE) を受信する。これによりノードAとEが同一レベルの障害を同時に検出していることを知り、ノードBにおいて障害が発生していることを検出する。障害発生ノードであるノードBを特定するために各ノードは予め当該論理リングに属する全ノードの番号を情報 (Ring Topology Map) として持っている。論理リング1のもつRing Topology Mapを図12に示す。これは論理リング1に属するノードの番号を接続順に記載したものである。ノードAはメッセージM1の優先度が同一でかつ発信ノード番号が宛であるということを検出し、該Ring Topology Mapを当該メッセージを受信した方向と反対方向 (本例の場合、反時計まわり) に自ノードを起点としてたどる。そしてノードEの番号が出現するまでに読み出されたノード番号 (この場合、ノードBのみ) が障害ノードであることを知る。

【0037】次にノードAは、障害ノード (ノードB) で終端あるいは他の論理リングへわたるパスを検出する。これには予めOSより配信されたNode Cross-Connect Mapを用いて行う。本実施例におけるNode Cross-Connect Mapを図13に示す。図13におけるNode Cross-Connect Mapには、論理リングに属する各論理VCの送信ノードと受信ノードの番号および該論理VCが他の論理リングからきた場合にはその論理リング番号と当該論理リング番号内における論理VCの番号を記す。また該論理VCが他の論理リングへ出力される場合にもその論理リング番号と当該論理リング番号内における論理VCの番号を記す。本実施例の場合には論理VCは1本だけであり、その論理VCはノードBでは他の論理リングへ接続されていない。したがって、ノードAはいままでノードB方向に送出していたトラヒックをノードDの方向の予備論理VCにブリッジする。すなわち、論理VC#1w、#2wを論理VC#1p、#2pにブリッジする。あわせてノードBの方向から受信していた論理VC#1w、#2wをノードDから受信する論理VC#1p、#2pにスイッチする。上記処理終了後、ノードAは、short path方向にM1=SF-R/B、M2=A/S/FERF、long path方向にM1=SF-R/B、M2=A/L/Br&Swを送出する。

【0038】また、ノードEにおいても同様にノードAの

送出したメッセージを受信し、ノードBの障害を検出する。そして、ブリッジとスイッチを実行し、short path方向にM1=SF-R/B、M2=E/S/FERF、long path方向にM1=SF-R/B、M2=E/L/Br&Swを送出する。

【0039】上記シグナリングにより、ノードAとEの間でメッセージを交換し、迂回路を形成する。図14に論理リング1における上記メッセージの伝達シーケンスを示す。また、図15に論理リング2におけるメッセージの伝達シーケンスを示す。形成された迂回路を用いて、前記パスはA-D-E-F-Iの経路を通過することになる。この経路のうち、A-D-E間は論理リング1の予備、E-F間は論理リング2の現用、F-I間は論理リング4の現用論理VCを利用する。

【0040】最後にノードEに障害が発生した場合の処理について述べる。信号の伝達経路は、この場合も図4に示すように設定されているものとする。本例においてノードEに障害が発生した場合、論理リング1においては、ノードBとDが障害を検出する。論理リング2においてはノードBとFが障害を検出する。論理リング3においては、ノードDとHが障害を検出する。論理リング4においてはノードFとHが障害を検出する。したがって、ノードEに障害が発生した場合、障害復旧要求を出すのは論理リング1では、障害端ノードBとDであり、論理リング2では、障害端ノードBとFである。論理リング3においては、ノードDとHであり、論理リング4においてはノードFとHである。

【0041】論理リング1では、ノードBとノードDがメッセージを交換し、ノードB-E間とD-E間のトラヒックに対する迂回路をB-A-D間の予備容量を用いて構成する。論理リング2では、ノードBとノードFがメッセージを交換し、ノードB-E間とF-E間のトラヒックに対する迂回路をB-C-F間の予備容量を用いて構成する。論理リング3では、ノードDとノードHがメッセージを交換し、ノードD-E間とH-E間のトラヒックに対する迂回路をD-G-F間の予備容量を用いて構成する。論理リング4では、ノードFとノードHがメッセージを交換し、ノードF-E間とH-E間のトラヒックに対する迂回路をF-I-H間の予備容量を用いて構成する。図4のパスの場合、ノードA-B-E間においては論理リング1に属している。しかし、これをB-A-D間の予備容量を使用したのでは、ノードAとノードIの間にパスを設定することはできない。したがって、ノードBにおいては、当該パスを論理リング1内の迂回路に折り返さずに、論理リング2の予備容量に接続する。障害復旧後の経路は図16に示すように、A-B-C-F-Iとなる。このうち、A-B間は現用VC、B-C-F間は予備VC、F-I間は現用VCを用いて伝送される。

【0042】上記処理における各ノードの動作について詳細に述べる。障害発生後、論理リング1においては、ノードBとDが障害を検出する。論理リング2においてはノードBとFが障害を検出する。以下、論理リング1にお

ける動作について述べる。

【0043】ノードDは障害を検出し、short path方向にM1=SF-R/E、M2=D/S/FERF、long path方向にM1=SF-R/E、M2=D/L/IDLEを送出する。また、ノードBも障害を検出し、short path方向にM1=SF-R/E、M2=B/S/FERF、long path方向にM1=SF-R/E、M2=B/L/IDLEを送出する。

【0044】long pathに存在するノードAは、ノードBとDから送出されたメッセージを受信する。そしてM1を調べ当該メッセージが自分宛でないのを検出し、それぞれ次段に中継する。また、論理リング1において、受信側予備論理VCも送信側予備論理VCに接続する。

【0045】次にノードBは、ノードDの送出したlong path方向からのメッセージ (M1=SF-R/E、M2=D/L/IDLE)を受信する。これによりノードBとDが同一レベルの障害を同時に検出していることを知り、ノードEにおいて障害が発生していることを検出する。ノードBは受信メッセージM1の優先度が同一でかつ発信ノード番号がDであるということを検出し、Ring Topology Mapを当該メッセージを受信した方向と反対方向 (本例の場合、反時計まわり) に自ノードを起点としてたどる。そしてノードDの番号が出現するまでに読み出されたノード番号 (この場合、ノードEのみ) が障害ノードであることを知る。

【0046】次にノードBは、障害ノード (ノードE) で終端あるいは他の論理リングへ接続されるパスを検出する。これには予めOSより配信されたNode Cross-Connect Mapを用いて行う。本実施例におけるNode Cross-Connect Mapを図13に示す。本実施例の場合には、ノードEにおいて論理リング2へ接続される論理VCが存在することがわかる。ノードBは論理リング2に接している。従って当該論理VCを論理リング2へ接続するため、ノードBにおいて、論理リング2内部の接続先の論理VC (本実施例の場合、論理VC ID4) にブリッジする。また、スイッチに関しても同様に行う。

【0047】上記処理終了後、ノードBは、short path方向にM1=SF-R/E、M2=B/S/FERF、long path方向にM1=SF-R/E、M2=B/L/Br&Swを送出する。

【0048】また、ノードDにおいても同様にノードBの送出したメッセージを受信し、ノードEの障害を検出する。そして、ブリッジとスイッチを実行し、short path方向にM1=SF-R/E、M2=D/S/FERF、long path方向にM1=SF-R/E、M2=D/L/Br&Swを送出する。論理リング1におけるメッセージの伝達シーケンスを図17に示す。

【0049】次に論理リング2における動作について述べる。論理リング2においても上記と同様の動作がノードBとノードFの間で発生する。

【0050】ノードFは障害を検出し、short path方向にM1=SF-R/E、M2=F/S/FERF、long path方向にM1=SF-R/E、M2=F/L/IDLEを送出する。また、ノードBも障害を検出し、short path方向にM1=SF-R/E、M2=B/S/FERF、long path方向にM1=SF-R/E、M2=B/L/IDLEを送出する。

【0051】long pathに存在するノードCは、ノードBとFから送出されたメッセージを受信する。そしてM1を調べ当該メッセージが自分宛でないのを検出し、それぞれ次段に中継する。また、論理リング2において、受信側予備論理VCも送信側予備論理VCに接続する。

【0052】次にノードFは、ノードBの送出したlong path方向からのメッセージ (M1=SF-R/E、M2=B/L/IDLE)を受信する。これによりノードBとFが同一レベルの障害を同時に検出していることを知り、ノードEにおいて障害が発生していることを検出する。ノードFは受信メッセージM1の優先度が同一でかつ発信ノード番号がBであるということを検出し、Ring Topology Mapを当該メッセージを受信した方向と反対方向 (本例の場合、反時計まわり) に自ノードをたどる。そしてノードBの番号が出現するまでに読み出されたノード番号 (この場合、ノードEのみ) が障害ノードであることを知る。

【0053】次にノードFは、障害ノード (ノードE) で終端あるいは他の論理リングへ接続されるパスを検出する。これには予めOSより配信されたNode Cross-Connect Mapを用いて行う。本実施例におけるNode Cross-Connect Mapを図18に示す。本実施例の場合には、ノードEにおいて論理リング1へ接続される論理VCが存在することがわかる。ノードFは論理リング1には隣接していない。従って当該論理VCを論理リング1へ接続するため、ノードFにおいて、論理リング2内部の予備論理VC (本実施例の場合、論理VC ID4) にブリッジする。また、スイッチに関しても同様に行う。

【0054】上記処理終了後、ノードFは、short path方向にM1=SF-R/E、M2=F/S/FERF、long path方向にM1=SF-R/E、M2=F/L/Br&Swを送出する。

【0055】また、ノードBにおいても同様にノードFの送出したメッセージを受信し、ノードEの障害を検出する。そして、ブリッジとスイッチを実行し、short path方向にM1=SF-R/E、M2=B/S/FERF、long path方向にM1=SF-R/E、M2=B/L/Br&Swを送出する。論理リング2におけるメッセージの伝達シーケンスを図19に示す。

【0056】形成された迂回路を用いて、前記パスはA-B-C-F-Iの経路を通過することになる。この経路のうち、A-B間は論理リング1の現用、B-C-F間は論理リング2の予備、F-I間は論理リング4の現用論理VCを利用する。

【0057】上記に示した各ノードの処理のフローチャートを図20に示す。図20において論理リングから切り離されるノードとは、ノード障害または複数回線障害によって論理リングに接続が不可能になったノードである。図16ではノードEが該当する。

【0058】本方式を用いるノードにおいては時分割スイッチのアドレスコントロールメモリ (ACM) を複数持つことにより、上記動作を迅速に行うことが可能である。すなわち、図21に示すようにACMを3つ持つ構造にし、501つを通常時用 (ACM16)、1つを単一回線障害用 (ACM1

7)、1つをノード障害用(ACM18)に割り当てる。そしてセクタ15を制御することによりデータメモリ11からVCを読み出すごとに読み出しに用いるACMを前記3つのACMから選択する。

【0059】ノードB-E間の回線にのみ障害が発生した場合におけるノードEの動作について説明する。通常時、セクタ15はACM16を常時選択している。障害発生後、障害区間を通過するパスに関係有る論理VC、即ち論理リング1と2に属する論理VCの読み出しには、ACM17を用いて読み出すことにより、論理リング1と2内の予備論理VCへ折り返す。そして、その他の影響を受けないVCに関しては、ACM16を用いて通常どおり読み出す。

【0060】次にノードBにおいて障害が発生した場合におけるノードEの動作について説明する。障害発生後、障害発生ノードから他論理リングに接続されるVCについては、ACM18を用いて読み出すことにより、他論理リングの予備論理VCへ折り返す。障害ノードを通過し、かつ、他の論理リングに障害ノードでは接続されないパスに属する論理VCの読み出しには、ACM17を用いることにより、それぞれの論理リング内の予備論理VCへ折り返す。そして、その他の影響を受けない論理VCに関しては、ACM16を用いて通常どおり読み出す。

【0061】本例では、M1、M2のノード番号表示に4ビットしか割り当てていないので、各論理リングに属するノード数は16以下となる。しかし、マルチフレーム等の手法を使用し、伝達バイトの容量を増やせば、上記ノード数の制限を取り除くことは可能である。

【0062】本例では、シグナリングにVC-4 POH内の75バイトを用いたが、Section Overhead内のDCC(Data Communication Channel)を用いても実現は可能である。

【0063】本例では、SF-Rの実施例のみを示したが、他の命令も容易に実行可能である。

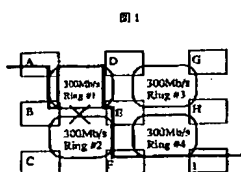
【0064】

【発明の効果】本発明によれば、ネットワークを複数の論理リングに分割し、迂回路の設定をおこなうため、障害復旧にかかる時間が小さくなるという効果がある。

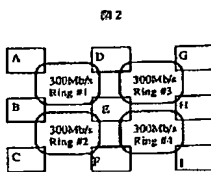
【図面の簡単な説明】

【図1】回線障害に対する復旧例を示す図。

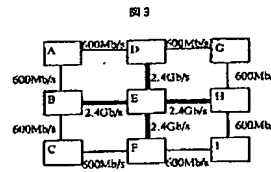
【図1】



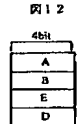
【図2】



【図3】



【図12】



【図2】本発明におけるリングの例を示す図。

【図3】本発明における物理網の例を示す図。

【図4】経路の例を示す図。

【図5】CCITT勧告に定めるSTM-4信号。

【図6】CCITT勧告に定めるSTM-16信号。

【図7】物理網と論理リングの対応。

【図8】M1とM2の内訳。

【図9】回線障害時の論理リング1におけるメッセージ伝達シーケンス。

10 【図10】回線障害時の論理リング2におけるメッセージ伝達シーケンス。

【図11】ノードB障害に対する復旧例を示す図。

【図12】論理リング1におけるRing Topology Map。

【図13】論理リング1におけるNode Cross-Connect Map。

【図14】ノードB障害時の論理リング1におけるメッセージ伝達シーケンス。

【図15】ノードB障害時の論理リング2におけるメッセージ伝達シーケンス。

20 【図16】ノードE障害に対する復旧例を示す図。

【図17】ノードE障害時の論理リング1におけるメッセージ伝達シーケンス。

【図18】論理リング2におけるNode Cross-Connect Map。

【図19】ノードE障害時の論理リング2におけるメッセージ伝達シーケンス。

【図20】各ノードの処理フローチャート。

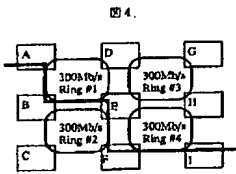
【図21】各ノードのスイッチ部の構成を示す図。

【符号の説明】

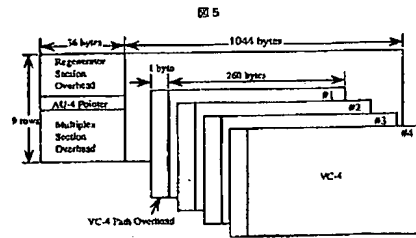
30 1~4…入力ハイウェイ、5~8…出力ハイウェイ、9~10…ハイウェイ、11…データメモリ、12…多重化部、13…分離部、14…書き込み制御部、15…セクタ、16~18…ACM、19…読み出し制御部、20…警報解読部、21…タイミング生成部、22…書き込みアドレス信号線、23…読み出しアドレス信号線、24…通常時アドレス、25…単一回線障害時アドレス、26…ノード障害時アドレス、27…ACMに与える読み出しアドレス、28…装置内クロック、29…警報情報。

(10)

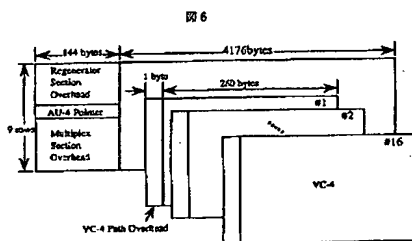
【図4】



【図5】



【図6】



【図7】

図7

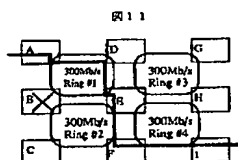
処理Ring #1	Node A-B	Node B-E	Node E-D	Node D-A
処理VC#1w	VC-4#1	VC-4#1	VC-4#1	VC-4#1
処理VC#2w	VC-4#2	VC-4#2	VC-4#2	VC-4#2
処理VC#1p	VC-4#3	VC-4#3	VC-4#3	VC-4#3
処理VC#2p	VC-4#4	VC-4#4	VC-4#4	VC-4#4
処理Ring #2	Node B-C	Node C-F	Node F-E	Node E-B
処理VC#1w	VC-4#1	VC-4#1	VC-4#1	VC-4#1
処理VC#2w	VC-4#2	VC-4#2	VC-4#2	VC-4#2
処理VC#1p	VC-4#3	VC-4#3	VC-4#3	VC-4#3
処理VC#2p	VC-4#4	VC-4#4	VC-4#4	VC-4#4
処理Ring #3	Node D-E	Node E-H	Node H-G	Node G-D
処理VC#1w	VC-4#5	VC-4#5	VC-4#5	VC-4#5
処理VC#2w	VC-4#6	VC-4#6	VC-4#6	VC-4#6
処理VC#1p	VC-4#7	VC-4#7	VC-4#7	VC-4#7
処理VC#2p	VC-4#8	VC-4#8	VC-4#8	VC-4#8
処理Ring #4	Node E-F	Node F-I	Node I-H	Node H-E
処理VC#1w	VC-4#5	VC-4#5	VC-4#5	VC-4#5
処理VC#2w	VC-4#6	VC-4#6	VC-4#6	VC-4#6
処理VC#1p	VC-4#7	VC-4#7	VC-4#7	VC-4#7
処理VC#2p	VC-4#8	VC-4#8	VC-4#8	VC-4#8

【図8】

図8

Byte M1(bits 1-4)			
bit 1	bit 2	bit 3	bit 4
Switch Priority(Bits 1-4)			
Byte M1(bits 5-8)			
bit 5	bit 6	bit 7	bit 8
Destination Node Identification			
Byte M2(bits 1-4)			
bit 1	bit 2	bit 3	bit 4
Source Node Identification (Bits 1-4)			
Byte M2(bits 5-8)			
bit 5	bit 6	bit 7	bit 8
Status (Bits 5-8)			

【図11】



【図13】

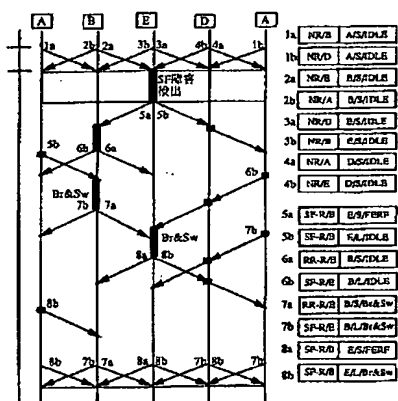
図13

接続先 処理Ring 内環路ノード番号	接続先 処理Ring 内環路ノード番号	接続先 処理Ring 内環路ノード番号	接続先 処理Ring 内環路ノード番号
処理VC#1	A	E	2/2
処理VC#2			

(11)

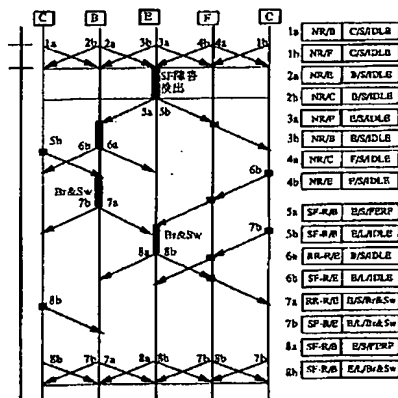
【図9】

図9



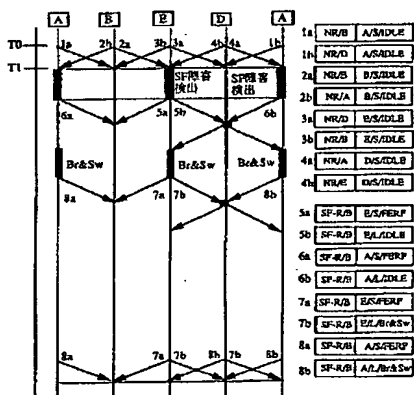
【図10】

図10



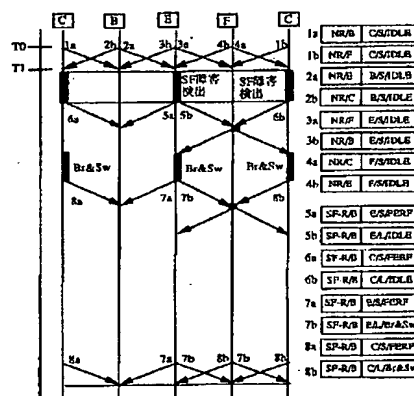
【図14】

図14

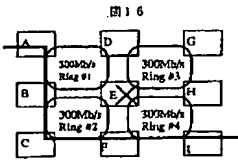


【図15】

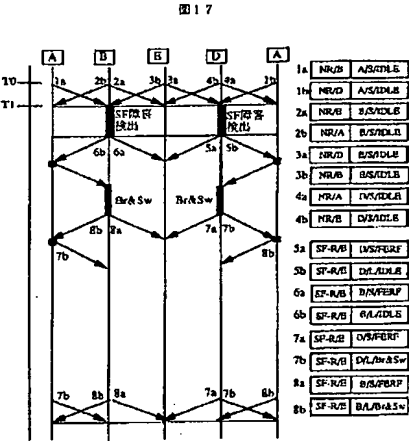
図15



【図16】



【図17】

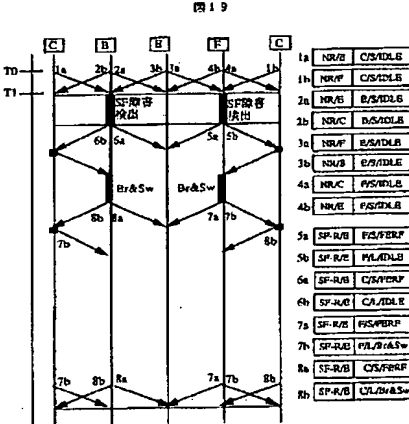


【図18】

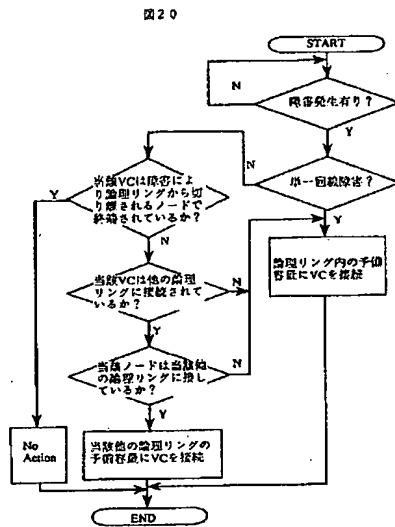
図18

接続先 送受信 VC番号	送受信 VC番号	送受信 VC番号	送受信 VC番号
送受信VC1			
送受信VC2	1/1	B	F

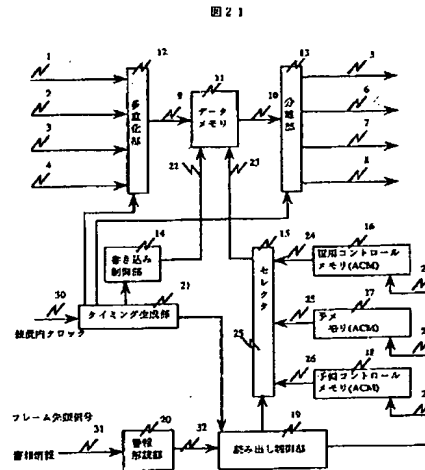
【図19】



【図20】



【図21】



フロントページの続き

(51)Int. Cl.⁶

H04L 12/26

29/14

識別記号

庁内整理番号

F I

技術表示箇所

8732-5K

H04L 11/08

9371-5K

13/00

311